Nivel de Red

Ruteo

http://www.bankroutingnumber.net/wp-content/uploads/2010/09/bank-routing-number-check2-e1285669232394.png

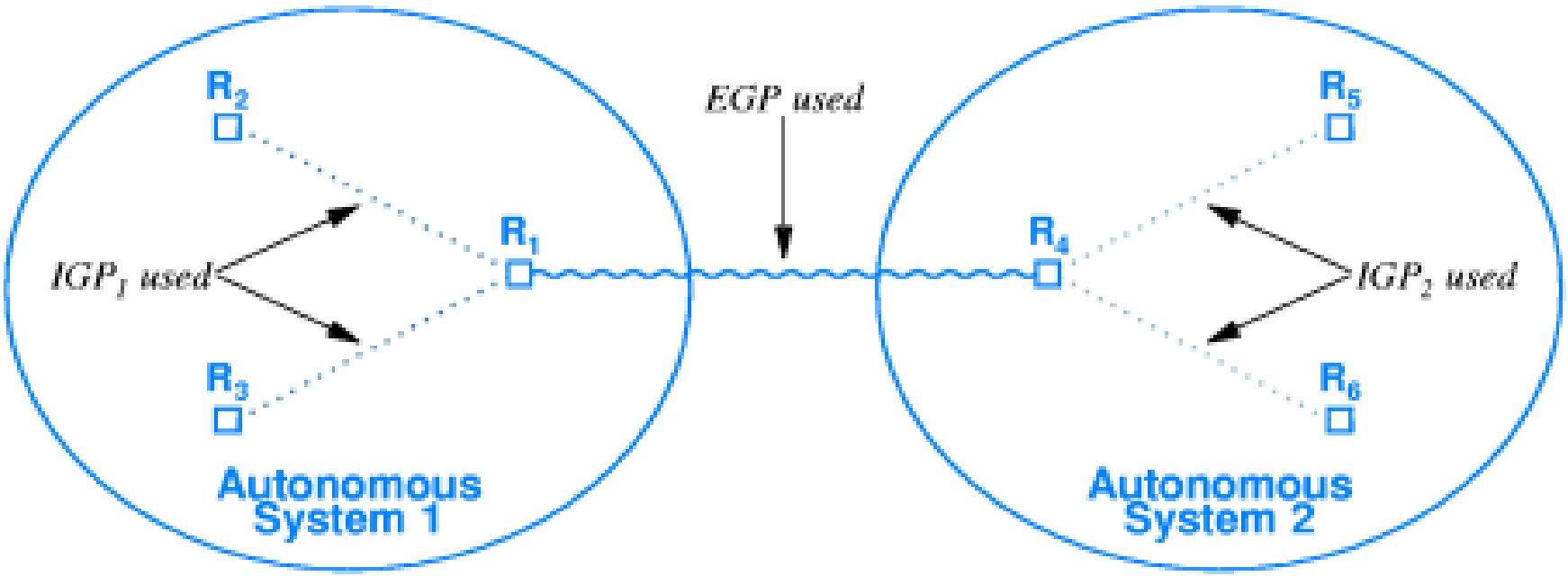
Agenda

- Introducción: ruteo interno y externo
- Algoritmos y protocolos
- Escalabilidad

Introducción

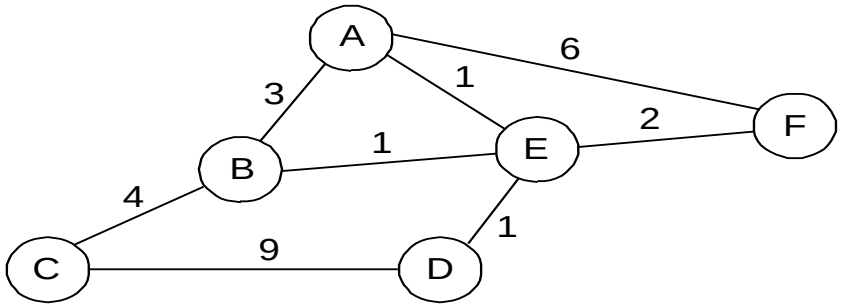
- Los protocolos de ruteo los podemos clasificar en :
  - Ruteo Interno (IGP , Internal Gateway Protocol)
  - Ruteo Externo (EGP, External Gateway Protocols)
- Los Internos su dominio de ruteo es un Sistema Autónomo (AS)
- Los externos son intradominios , es decir rutean entres AS
- Podemos decir que Internet es una interconexión de AS
- Lo anterior define un grafo o " red ", Internet existen varios "planos ", la red de routers , P2P , el grafo de WWW etc …….

Sistemas Autónomos (!!!)



Introducción

- Re-envío (forwarding) versus Ruteo
  - Re-envío: debe seleccionar una puerta de salida basado en la dirección destino y las tablas de ruteo.
  - ruteo: proceso mediante el cual las tablas de ruteo son construidas.
- Vemos la red como un Grafo



- Problema: Encontrar el camino de menor costo entre dos nodos
- Para la red anterior tener ruteo:
  - estáticos: topología
  - dinámicos: carga de los nodos y enlaces (Distribuidos y Centralizados)

Routing

(a)

Prefix/Length	Next Hop
18/8	171.69.245.10

(b)

Prefix/Length	Interface	MAC Address
18/8	if0	8:0:2b:e4:b:1:2

(a) Tabla de routing

(b) Tabla de forwarding

Protocolos de Ruteo Interno

RIP

OSPF

Vector de distancia vs Estado del enlace

	DISTANCE-VECTOR	LINK-STATE
Qué informa cada nodo?	* Su Tabla de Ruteo	* Estado de sus Enlaces
A quién pasa la infomación?	* Sólo a sus vecinos	* Inunda a toda la red
Algoritmo utilizado	* Bellman-Ford Distribuido	* Dijkstra
Datos utilizados	* Información de los vecinos	* Estado de Enlaces de cada nodo
Estructuras de Datos	* Tabla de Distancias * Tabla de Ruteo	* Tabla de Estado de Enlaces * Tabla de Ruteo
Caracteríticas	* Ciclos de Ruteo * Gran variedad de Algoritmos: * Merlin-Segall * Jaffe-Moss * Esquema OP * Diffusing Comp * Cheng * Cálculo Distribuido	* Visión Consistente de la Red * Gran uso de CPU y Memoria * Algoritmo Básico único * Cálculo Centralizado
Ejemplo de Protocolos de Internet	* RIP	* OSPF

Fuente :"Un nuevo protocolo de ruteo Interno" Lijding – Righetti 1997

Protocolos ruteo interno mas populares

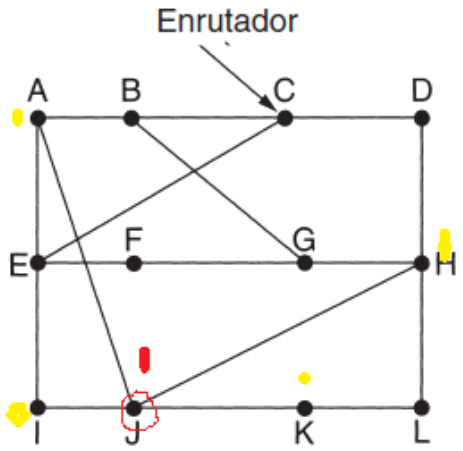
- RIP: Routing Information Protocol
  - desarrollado por XNS
  - Distribuido con Unix
  - usa algoritmo vector distancia
  - basado en cuenta de hops
- OSPF: Open Shortest Path First
  - Muchos textos dicen "reciente estándar en Internet "(!!!!!)
  - usa el algoritmo de estado de enlaces
  - soporta balanceo de carga y QoS (??)
  - soporta autenticación

Distance Vector

Vector de Distancia

- Cada nodo mantiene
  - `(Destination, Cost, NextHop)`
- Intercambia actualizaciones con sus vecinos directamente conectados
  - Periódicamente (en el orden de varios segundos)
  - Cuando su tabla cambia (se habla de una actualización gatillada)
- Cada actualización es una lista de pares:
  - `(Destination, Cost)`
- Se actualiza tabla local si se recibe una mejor ruta
  - Costo menor
  - llegó desde el host próximo "next-hop"
- Se refrescan rutas existentes; se borran si hay time out

Vector de distancia (I ejemplo)



(a)

	A	I	H	K	Nuevo retardo estimado desde J	Línea
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	–
K	24	22	22	0	6	K
L	29	33	9	9	15	K

Retardo JA es de 8 | Retardo JI es de 10 | Retardo JH es de 12 | Retardo JK es de 6

Vectores recibidos de los cuatro vecinos de J

Nueva tabla de enrutamiento para J

(b)

(a) Subred. (b) Entrada de A, I, H, K y la nueva tabla de enrutamiento de J.

- Este ejemplo se tomo "Redes de Ordenadores" Tanenbaum 4 edición, asume que estamos en "régimen" y los costos son delays

La tabla de ruteo

- las primeras cuatro columnas de la parte (b) vectores de retardo recibidos de los vecinos del router J .*A indica tener un retardo de 12 mseg a B, un retardo de 25 mseg a C, un* retardo de 40 mseg a *D, etc.*
- *J ha medido o estimado el retardo a sus vecinos A, I, H y K en 8, 10, 12 y 6 mseg, respectivamente*
- *J calcula su nueva ruta al router G. Sabe que puede llegar a A* en 8 mseg, y *A indica ser capaz de llegar a G en 18 mseg, por lo que J sabe que puede contar con* un retardo de 26 mseg a *G si reenvía a través de A los paquetes destinados a G.*
- *J calcula el retardo a G a través de I, H y K en 41 (31 + 10), 18 (6 + 12) y 37 (31 + 6) mseg,* respectivamente. El mejor de estos valores es el 18, por lo que escribe una entrada en su tabla deenrutamiento indicando que el retardo a *G es de 18 mseg, y que la ruta que se utilizará es vía H.*

Vector de Distancia (II ejemplo)

- Cada nodo arma un vector que contiene la distancia (costos) a todos los nodos y lo envía a sus vecinos

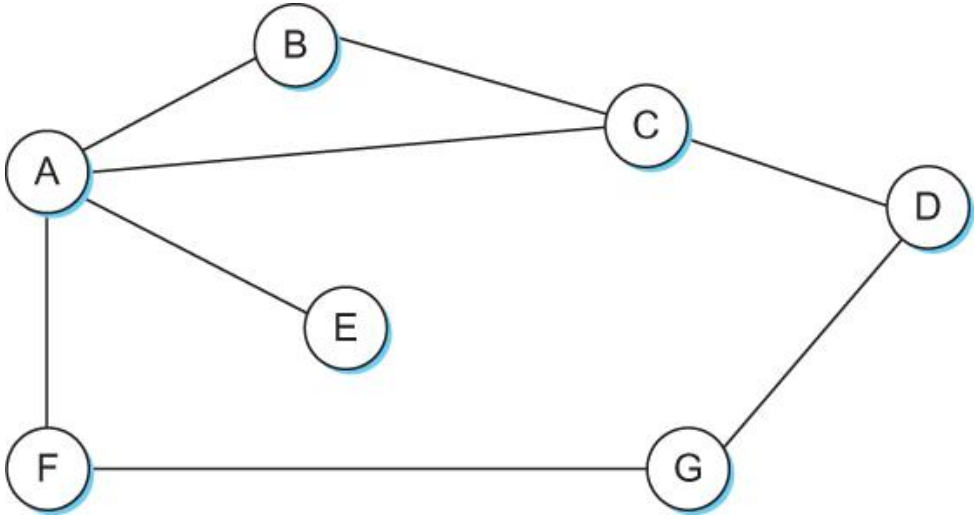


Tabla de ruteo del nodo A

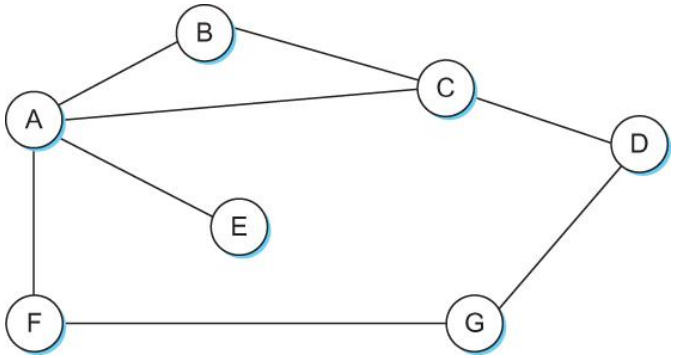
Destino	Costo	NextHop
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	2	F

Distance Vector (II)

Information Stored at Node	Distance to Reach Node						
	A	B	C	D	E	F	G
A	0	1	1	∞	1	1	∞
B	1	0	1	∞	∞	∞	∞
C	1	1	0	1	∞	∞	∞
D	∞	∞	1	0	∞	∞	1
E	1	∞	∞	∞	0	∞	∞
F	1	∞	∞	∞	∞	0	1
G	∞	∞	∞	1	∞	1	0

Estado de distancia inicial alamacenado en cada nodo (una visión global)

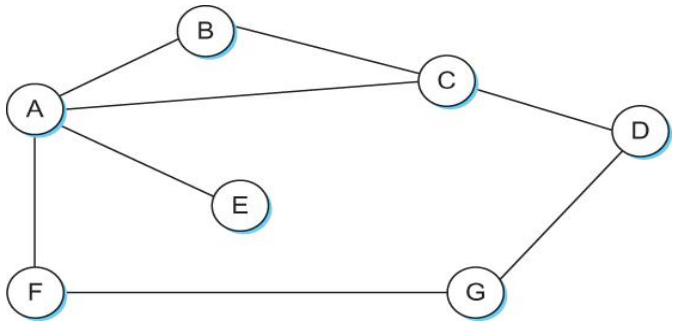
Distance Vector (II)



Destination	Cost	NextHop
B	1	B
C	1	C
D	∞	—
E	1	E
F	1	F
G	∞	—

Table de ruteo inicial de nodo A

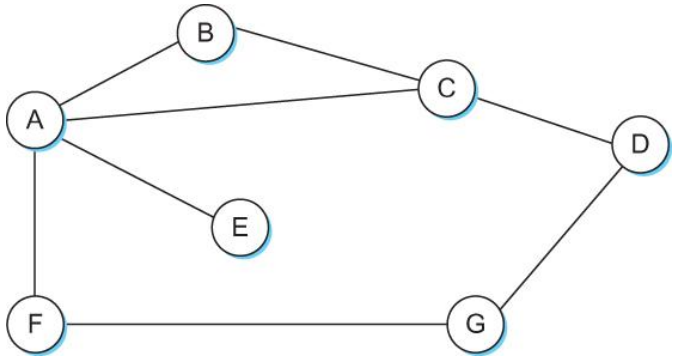
Distance Vector (II)



Destination	Cost	NextHop
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	2	F

Tabla de ruteo final de A

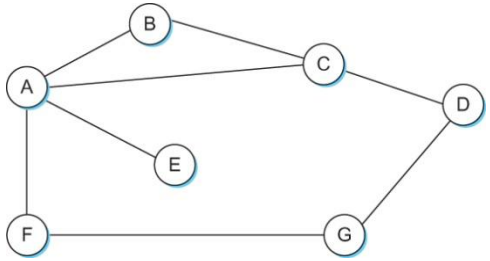
Distance Vector (II) Tabla global converge a :



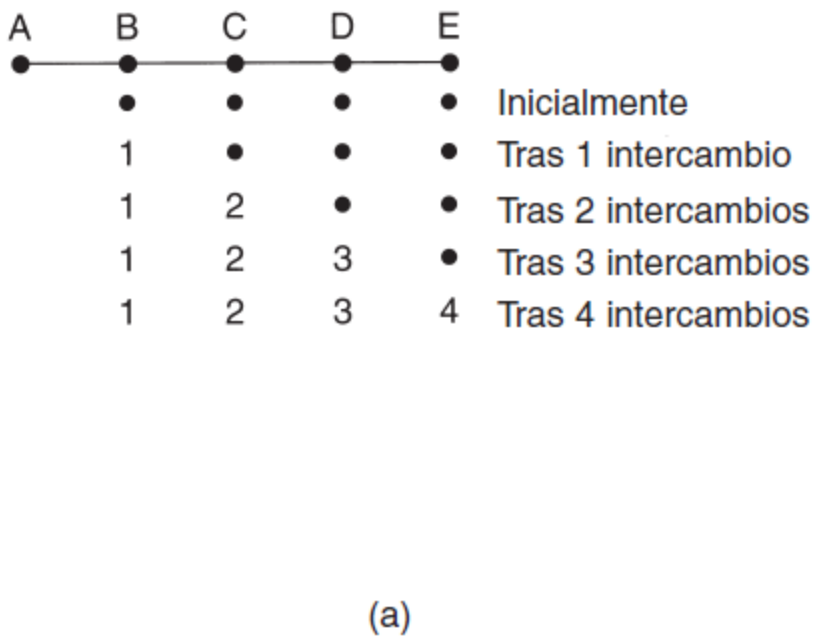
Information Stored at Node	Distance to Reach Node						
	A	B	C	D	E	F	G
A	0	1	1	2	1	1	2
B	1	0	1	2	2	2	3
C	1	1	0	1	2	2	2
D	2	2	1	0	3	2	1
E	1	2	2	3	0	2	3
F	1	2	2	2	2	0	1
G	2	3	2	1	3	1	0

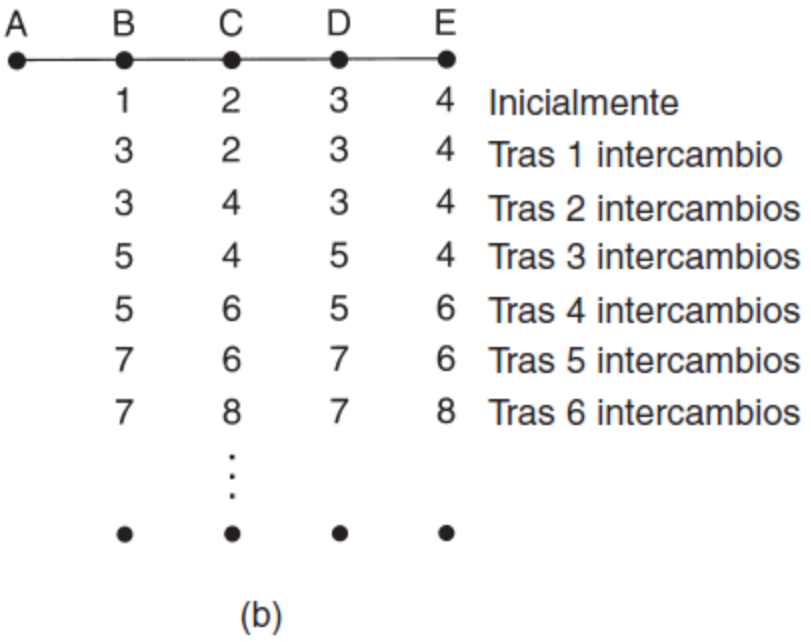
Ciclos de actualización

- Ejemplo 1
  - F detecta que enlace a G ha fallado
  - F fija distancia a G infinita y envía una actualización a A
  - A fija distancia a G infinita porque A usa F para llegar a G
  - A recibe actualización periódica de C con camino de 2 hops a G
  - A fija distancia a G como 3 y envía actualización a F
  - F decide él puede llegar a G en 4 hops vía A
- Ejemplo 2
  - Enlace de A a E falla.
  - A comunica distancia ∞ a E
  - B y C comunican distancia 2 a E
  - B decide que puede llegar a E en 3 hops; comunica esto a A
  - A decide que puede llegar a E en 4 hops; comunica esto a C
  - C decide que puede llegar a E en 5 hops...



Problema de conteo a Infinito



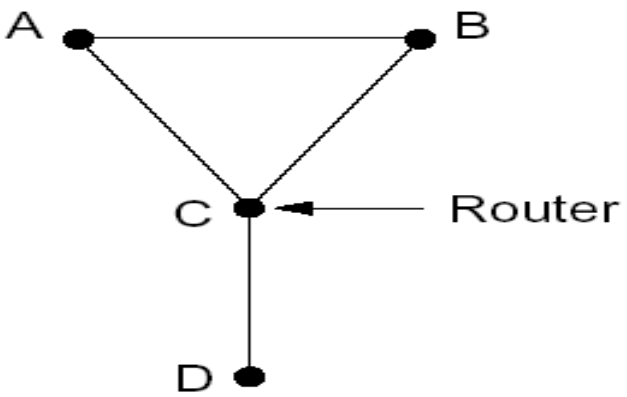


(a) El nodo A , esta seteado como infinito , luego se activa
(b) El nodo A , esta activo , luego cae

Heurísticas para romper los ciclos

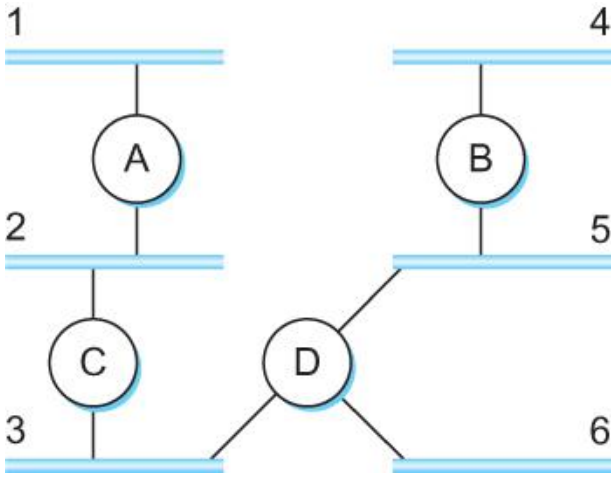
- Fijar 16 como infinito, es decir cuando el costo llega a 16 se asume que no hay ruta al nodo
- Partir el horizonte (Split horizon): omite la información de distancia que fue obtenida del nodo al cual se le envía el vector.
- Partir el horizonte con reverso venenoso (Split horizon with poison reverse): incluye las entradas obtenidas desde el nodo al cual se envía el vector pero esos destinos se les pone costo infinito.
- Las últimas dos técnicas sólo operan cuando el lazo involucra dos nodos.
- La convergencia de este protocolo no es buena, se mejora con ruteo usando el estado de los enlaces.

Ejemplo con Poison Reverse :

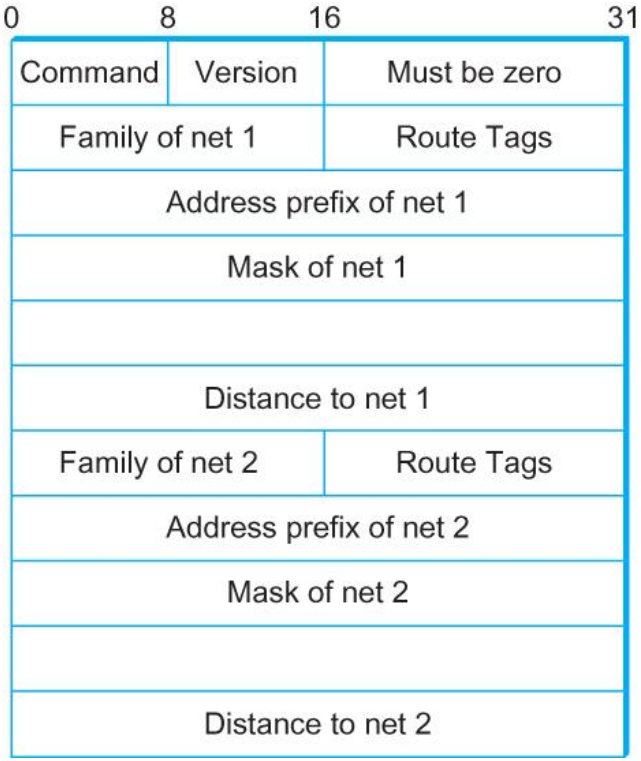


(a) Todos los costos son uno
(b) Falla el enlace C –D
(c) A puede llegar a D por B
(d) ……

Routing Information Protocol (RIP)



Network con RIP



Formato del paquete RIPv2

Link State

Algoritmo estado del enlace

- La topología de red y costos conocidos por todos los nodos
- “link state broadcast”: Todos los nodos tienen la misma información
- Computo del camino mínimo: Dijkstra

(forward search)

Estado del Enlace

- Estrategia
  - Enviar a todos los nodos (no sólo los vecinos) información sobre enlaces directamente conectados (no la tabla completa) "se *inunda*"

- Paquete del estado del enlace (Link State Packet,LSP)
  - id del node que creó el LSP
  - costo del enlace a cada vecino directamente conectado
  - número de secuencia (SEQNO)
  - time-to-live (TTL) para este paquete

Estado del Enlace (cont)

- Inundación Confiable
  - almacena el LSP más reciente de cada nodo
  - re-envía LSP a todos excepto a quien me lo envió
  - genera un nuevo LSP periódicamente
    - incrementa SEQNO
  - inicia SEQNO en 0 cuando reboot
  - decrementa TTL de cada LSP almacenado
    - descarta cuando TTL=0

Cálculo de Ruta

- Algoritmo de Dijkstra para el camino más corto entre nodos
- Sea
  - *N* denota el conjunto de nodos del grafo
  - *l* (*i*, *j*) denota costo no negativo (peso) para arco (*i*, *j*). Si no hay arco, el costo es infinito.
  - *s* denota este nodo
  - *M* denota el conjunto de nodos incorporados hasta ahora
  - *C*(*n*) denota el costo del camino de *s* a *n*

```
M = {s}
for each n in N - {s}
  C(n) = l(s, n)
while (N != M)
  M = M unión {w} tal que C(w) es el mínimo para todo w
  en (N - M)
  for each n in (N - M)
    C(n) = MIN(C(n), C (w) + l(w, n ))
```

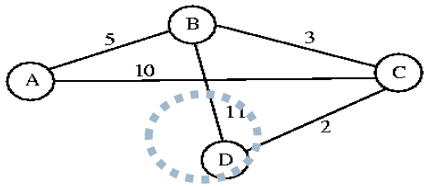
“Shortest Path Routing”

- En la práctica el cálculo de la tabla de ruta se hace conforme los LSP (link-state packet) llegan, utilizando una implementación de algoritmo de Dijkstra llamada “*forward search algorithm*”

- Se manejan dos listas la de entradas ***tentativas*** y las ***confirmadas***.

- Cada una se esa lista entradas : (Destination, Cost, NextHop)

Uso del algoritmo de Dijkstra en ruteo



Paso	Confirmado	Tentativo
1	(D,0,-)	
2	(D,0,-)	(B,11,B)
		(C,2,C)
3	(D,0,-)	(B,11,B)
	(C,2,C)	
4	(D,0,-)	(B,5,C)
	(C,2,C)	(A,12,C)
5	(D,0,-)	(A,12,C)
	(C,2,C)	
	(B,5,C)	
6	(D,0,-)	(A,10,C)
	(C,2,C)	
	(B,5,C)	

Comentario

Miramos el LSP de D porque es el único nuevo miembro confirmado

El LSP de D nos dice cómo llegamos a B y C

El miembro de menor costo entre los tentativos es C. Se examina el LSP del nodo recién confirmado

Con C, se actualizan los costos. Ahora llegamos a B vía C y se incorpora A

Se mueve el de menor costo de tentativos a confirmados.

Con B, se actualiza los costos. Ahora se llega a A vía B.

Después de esto se mueve el menor costo de los tentativos a los confirmados. A es el único y terminamos.

Comparación entre algoritmo vector de distancia y estado de enlaces.

- En el algoritmo vector de distancia cada nodo transmite a sus vecinos lo que sabe respecto de toda la red (distancia a todos los nodos).
- En el algoritmo estado de enlaces cada nodo transmite a toda la red lo que sabe de sus vecinos. (el estado de sus vecinos)
- El segundo es **estable**, no genera gran tráfico y responde rápido a cambios de topología.
- El problema del segundo es la cantidad de información almacenada en los nodos(un LSP por nodo)

Métricas

- Métrica ARPANET original
  - mide el número de paquetes encolados en cada enlace
  - No toma en cuenta latencia ni ancho de banda
- Métrica ARPANET nueva
  - Marca cada paquete entrante con su tiempo de llegada (arrival time, **`AT`**)
  - Graba tiempo de salida (departure time, **`DT`**)
  - Cuando llega el ACK del enlace de datos llega, calcula

    `Delay = (DT - AT) + Transmit + Latency`
    `Transmit y Latency son parámetros estáticos del enlace.`
  - Si hay timeout, reset **`DT`** a tiempo de salida para retransmisión
  - Costo del enlace = retardo promedio medido en algún periodo
- Mejora fina
  - Reducir el rango dinámico para el costo
  - En lugar de retardo se emplea la utilización del enlace

OSPF

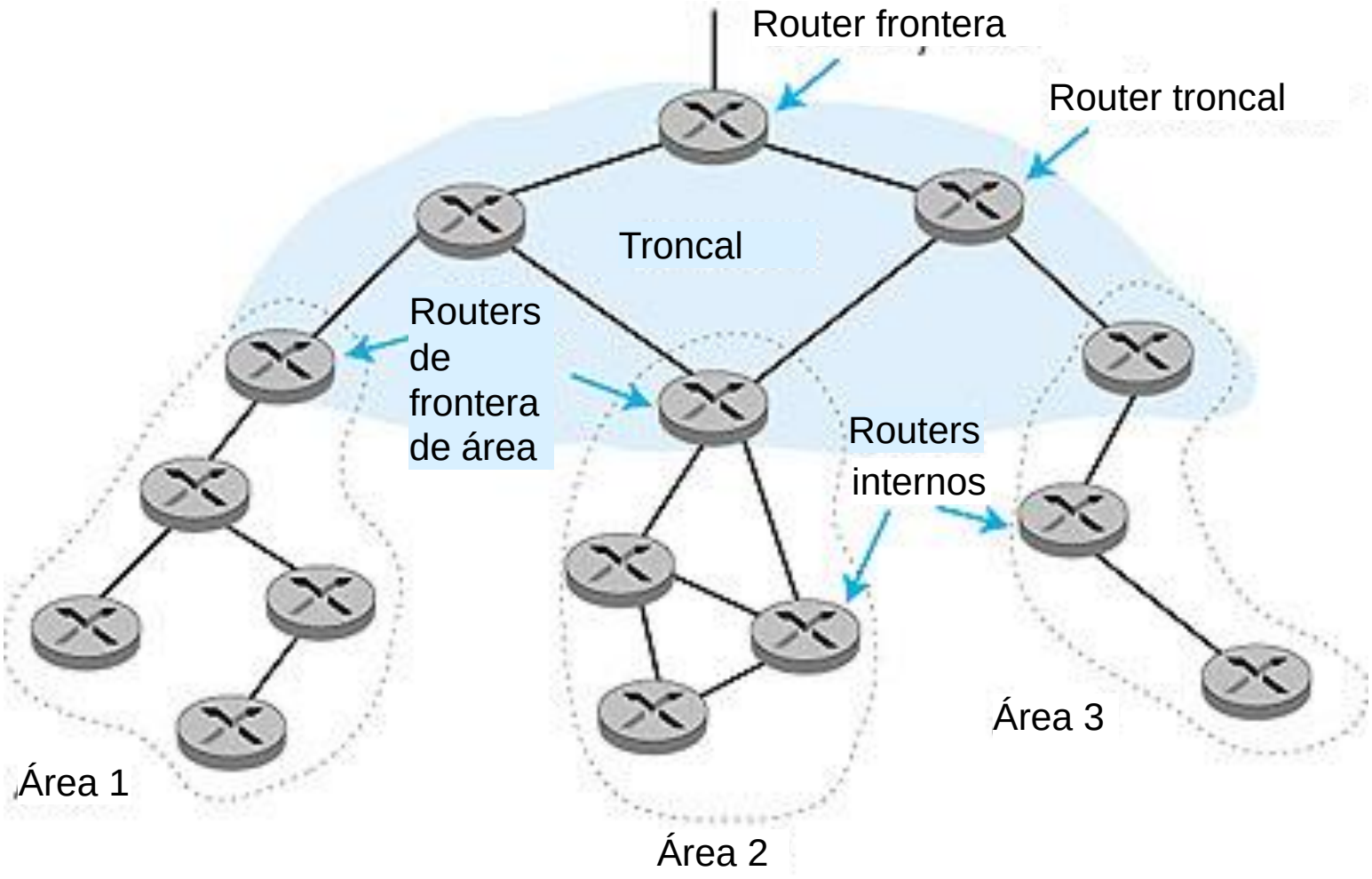
- "Abierto": disponible públicamente.
- Utiliza un algoritmo de estado de enlaces:
  - Distribución de paquetes del estado de enlaces.
  - Mapa de la topología en cada nodo.
  - Cómputo de la ruta usando el algoritmo de Dijkstra.

- El anuncio OSPF lleva una entrada por cada router vecino.
- Anuncios distribuidos a todos los SA (mediante inundación):
  - Transportados por mensajes en OSPF directamente sobre IP (mejor que TCP o UDP).

OSPF

- Seguridad: todos los mensajes OSPF están autentificados (para prevenir intrusiones malignas).
- Múltiples caminos con el mismo coste permitidos (sólo un camino en RIP).
- Para cada enlace, varias métricas de coste para distintos TOS (por ejemplo, coste del enlace satélite "rebajado" para un mayor rendimiento; alto para tiempo real) (en teoría …)
- Soporte integrado uni- y multidifusión:
  - La multidifusión OSPF (MOSPF) usa la misma base de datos topológica que OSPF.
- OSPF jerárquico en grandes dominios (escala).

OSPF jerárquico



OSPF jerárquico

- Dos niveles de jerarquía: área local, troncal.
  - Anuncios de estado de enlace sólo en el área.
  - cada nodo detalla la topología del área; sólo conoce la dirección (el camino más corto) a las redes de otras áreas.
- Routers de frontera de área: "resumen" las distancias a las redes del mismo área, anuncian a otros routers de Frontera de Área.
- Routers troncales: ejecutan ruteados de OSPF limitados al troncal.
- Routers frontera: conectan con otros SAs.

Tipos de Mensaje en OSPF

Tipo de mensaje	Descripción
Hello	Descubre quiénes son los vecinos
Link state update	Proporciona los costos del emisor a sus vecinos
Link state ack	Confirma la recepción de la actualización del estado del enlace
Database description	Anuncia qué actualizaciones tiene el emisor
Link state request	Solicita información del socio

Tipos de Mensaje en OSPF

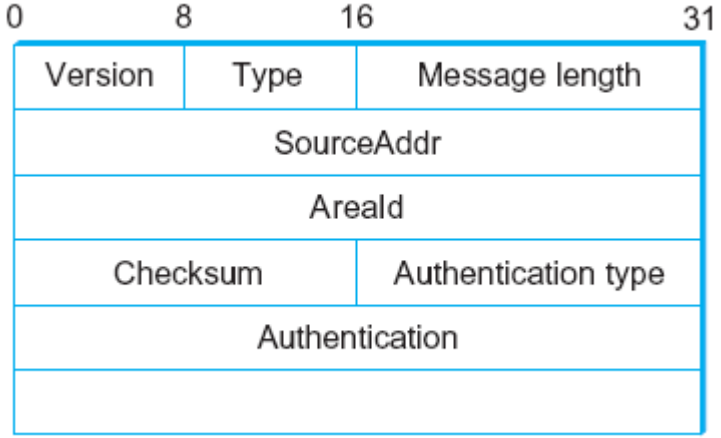
- Cada router inunda periódicamente con mensajes LINK STATE UPDATE a cada uno de routers adyacentes. Este mensaje da su estado y proporciona los costos usados en la base de datos topológica. Para hacerlos confiables, se confirma la recepción de los mensajes de inundación. Cada mensaje tiene un número de secuencia para que un enrutador pueda ver si un LINK STATE UPDATE entrante es más viejo o más nuevo que el que tiene actualmente. Los routers también envían estos mensajes cuando una línea su costo cambia.

- Los mensajes DATABASE DESCRIPTION dan los números de secuencia de todas las entradas de estado del enlace poseídas por el emisor actualmente. Comparando sus propios valores con los del emisor, el receptor puede determinar quién tiene los valores más recientes. Estos mensajes se usan cuando se activa una línea.
- Cualquier socio puede pedir información del estado del enlace al otro usando los mensajes LINK STATE REQUEST.
- El resultado de este algoritmo es que cada par de enrutadores adyacentes hace una verificación para ver quién tiene los datos más recientes, y de esta manera se difunde la nueva información a lo largo del área.

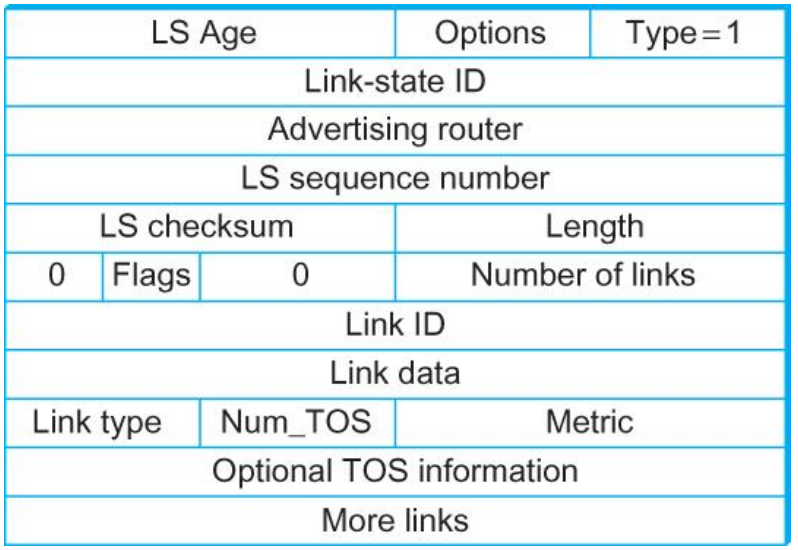
Dos niveles de jerarquía: área local, troncal.

- Anuncios de estado de enlace sólo en el área.
- cada nodo detalla la topología del área; sólo conoce la dirección (el camino más corto) a las redes de otras áreas

Open Shortest Path First (OSPF)



Formato header OSPF



Formato de "OSPF Link State Advertisement" (LSA)

BGP

EGP: Exterior Gateway Protocol

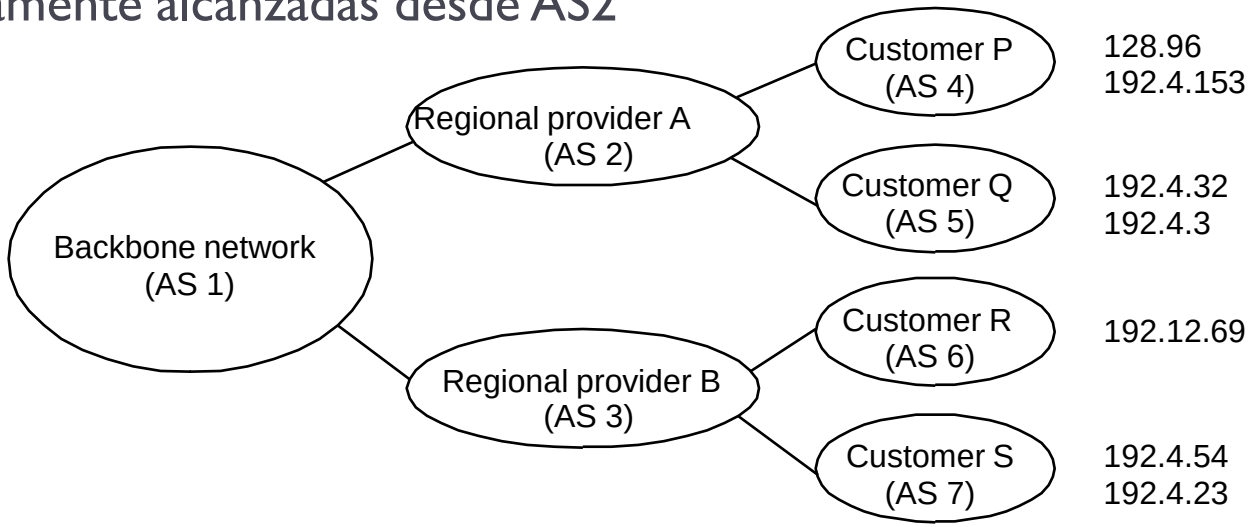
- Generalidades
  - diseñado para una Internet estructurada como árbol
  - se preocupa de *alcanzar* los nodos, no optimiza rutas
- Mensajes del Protocolo
  - Adquisición de vecinos: un router requiere que otro sea su par; pares intercambian información de alcance
  - Alcance de vecinos: un router periódicamente prueba si el otro es aún alcanzable; intercambia mensajes HELLO/ACK;
  - actualización de rutas: pares periódicamente intercambian sus tablas de ruteo (vector distancia)

BGP-4: Border Gateway Protocol

- Tipos AS
  - stub AS: tiene una única conexión a otro AS
    - transporta sólo tráfico local
  - multihomed AS: tiene conexiones a más de un AS
    - no transporta tráfico en transito
  - transit AS: tiene conexiones a más de un AS
    - transporta ambos tráfico local y en transito
- Cada AS tiene:
  - Uno o más routers de borde
  - Un "portavoz " BGP que publica:
    - Redes locales
    - Otras redes alcanzables (sólo el transit AS)
    - entrega información de rutas (*path)*

Ejemplo: BGP

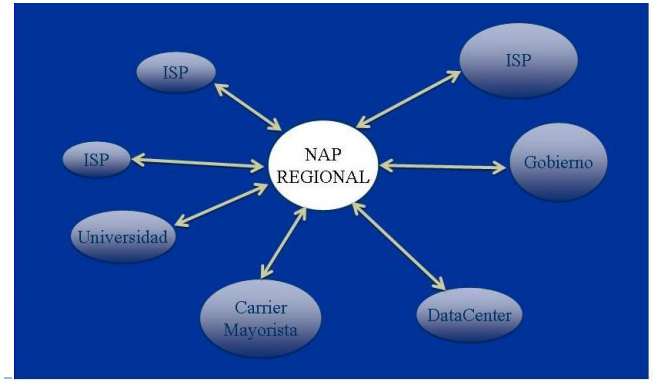
- Portavoz para AS2 publica alcanzabilidad a P y Q
  - redes 128.96, 192.4.153, 192.4.32, y 192.4.3, pueden ser directamente alcanzadas desde AS2



- Portavoz para backbone publica
  - redes 128.96, 192.4.153, 192.4.32, y 192.4.3 pueden ser alcanzadas a lo largo de la ruta (AS1, AS2).
- Portavoz puede cancelar una ruta publicada previamente

BGP : Los NAPs ….

- "Los NAPs (Network Access Points por sus siglas en inglés) o también conocidos como IXs (Internet eXxchanges) son componentes fundamentales de la Red Internet. Através de un NAP, se produce el intercambio de tráfico entre las redes de diversas entidades (operadores, proveedores de acceso, organismos de gobierno, entidades académicas, etc.) Estos puntos neurálgicos de la Red se han construido en todo el mundo bajo distintos esquemas institucionales, topológicos y operacionales. No obstante, la mayoría de ellos persigue idénticos objetivos: eficientizar el ruteo de Internet, mejorando la calidad de servicio y minimizar los costos de interconexión. Todos los NAPs CABASE siguen el modelo cooperativo. Todos los miembros de los NAPs CABASE, son socios de la Cámara Argentina de Internet que tienen como objetivo mejorar la calidad en las comunicaciones y reducir costos. Generalidades"[1]



[1] http://www.cabase.org.ar/wordpress/que-es-un-nap/

Algunas Referencias

- RIPV1
  - RFC-1058 - Charles Hedrick, June 1988
- RIPV2
  - RFC 1388 (Jan. 1993), RFC 1723 (Nov. 1994), RFC 2453 (Nov. 1998),, Gary Malkin
- OSPFV1
  - RFC 1131 J. Moy, Oct. 1989
- OSPFV2
  - RFC 1247 (July 1991), RFC 1583 (March 1994), RFC 2178 (July 1997), RFC 2328 (April 1998), J. Moy